AD-A281 544

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 6/14/94	3. REPORT TYPE AND DATES COVERED Final 28 Sep 92 - 27 Dec 93

4. TITLE AND SUBTITLE
Real-Time Simulation of Large-Scale Multibody Systems using Automated Equation Decoupling Techniques

5. FUNDING NUMBERS
DAAL03-92-G-0409

6. AUTHOR(S)
Linda R. Petzold
Roger A. Wehage

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Computer Science Department
University of Minnesota
4-192 EE/CS Bldg
200 Union St SE
Minneapolis, MN 55455

U.S. Army TARDEC
Attn: AMSTA-RY
Warren, MI 48397-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U. S. Army Research Office
P. O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
ARO 30980.1-MA

11. SUPPLEMENTARY NOTES
The view, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
A method for formulating highly optimized vehicle models using symbolic analysis has been developed. The vehicle model makes extensive use of computations which can be precomputed in advance of the real-time simulation. The new approach has been used at TARDEC for the efficient simulation of several military vehicles.

14. SUBJECT TERMS symbolic analysis, multibody systems, precomputed coefficients, vehicle simulation	15. NUMBER OF PAGES 11
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Final Report: Real-Time Simulation of Large-Scale Multibody Systems using Automated Equation Decoupling Techniques

Linda R. Petzold
Computer Science Department
University of Minnesota
Minneapolis, MN 55455

Roger A. Wehage
U.S. Army TARDEC
ATTN: AMSTA-RY
Warren, MI 48397-5000

1. INTRODUCTION

The Army routinely employs high-resolution, computer-based vehicle models to help set performance specifications and identify upper bounds on safe operation. Validated model predictions are used to support the development, product improvement, deployment and training processes. Computer-aided vehicle analysis is also important because changing battlefield requirements and emerging technology are pushing designs toward increased complexity. For example, the operator's influence on system performance must be accurately represented because current designs are placing more emphasis on systems that adapt to his capabilities. Therefore models must be capable of incorporating operator inputs in real time when they are important. Vehicles also have critical subsystems that must be accurately modeled or represented empirically, and optimized to run in real time along with the equations of motion. Effective modeling and simulation requires (1) accurate characterization of vehicle systems, (2) conversion of data into models, and (3) ease of application to diverse problems.

Computational dynamics, as we know it today, began with the development of analog and digital computers in the 1950s and 1960s[1], but the theory of constrained multibody dynamics was developed in the late 1600's to the 1800s[2,3]. Initial vehicle models were necessarily simplified and hand optimized to fit existing architectures, and they evolved in the 1960's as newer computers were developed. At that time, a single model may have required two or more years of concentrated developmental effort. The trend in the 1970's and early 1980's was toward general purpose programs to provide simpler environments for rapid model development. These programs exploited the latest advances in dynamics algorithms, numerical methods and compiler design, but their generality made them much less efficient. In the 1980's, novel algorithms, combined with vector and parallel processors, were invented to control robotic devices.

Researchers also explored symbolic processors to augment or replace hand manipulation and coding. Now, symbolic processing, with emphasis on fitting computational algorithms to unique multiprocessor hardware and software architectures, has become commonplace when developing vehicle dynamics models. However, many of the computationally intensive procedures associated with general methods were retained, and the solution to these residual inefficiencies was to employ more powerful and expensive computers.

Most vehicle system models used for real-time applications have some nice features that can be exploited to gain computational advantage over general simulation methods. With better understanding of these features, a number of novel ad hoc techniques can be used to reduce computational overhead, often by orders of magnitude. However, most of them are difficult to implement, and they require more expertise and development time to obtain working models. But they do have the advantage of allowing larger, more detailed real-time models to be executed on less expensive computers.

2. TRADITIONAL APPROACHES TO FORMULATING EQUATIONS OF MOTION

To better understand how computational overhead may be reduced, it is useful to briefly illustrate a vehicle's kinematic, dynamic and kinetic equation structures. This can be done in a number of ways, but an approach which exploits graph theoretic descriptions of topology and object oriented representations using spatial algebra provides more direct paths to the desired results. The ultimate goal is to minimize run-time overhead by relegating many of the operations to preprocessors and avoiding others entirely.

Numerous multibody formalisms exist in the literature and can be classified roughly into three categories[4] as described below. Newton first defined equations for the translational motion of a particle and Euler developed them for the rotational motion of a rigid body. Together, Newton-Euler equations[2] (*NEE*) give the absolute spatial motion of rigid bodies in a Cartesian setting. Joints are represented by imposing algebraic constraints on the displacements, and by appending second time derivatives of these equations to the *NEE* using Lagrange multipliers. Many modern computer algorithms are based on *NEE* with appended constraints[4]. The number of variables in models based on this method can be large and the constrained equations will generally have the factored block matrix form

$$\begin{bmatrix} M_a & J_a^T \\ J_a & 0 \end{bmatrix} \begin{bmatrix} a \\ -\lambda \end{bmatrix} = \begin{bmatrix} g \\ \gamma \end{bmatrix} \tag{1}$$

Lagrange developed a second approach based on kinetic energy expressed in terms of joint relative velocities. He defined a procedure to expand a kinetic energy expression into Lagrange's equations[2] (LE) and used multipliers to append constraint equations to obtain

$$\begin{bmatrix} M_p & J_p^T \\ J_p & 0 \end{bmatrix} \begin{bmatrix} \ddot{p} \\ -\lambda \end{bmatrix} = \begin{bmatrix} Q_p \\ \gamma \end{bmatrix} \tag{2}$$

Later, others such as Appell, Gibbs[2] and Kane[5] developed procedures to derive equations of motion with no appended constraint equations. Their formulations, which are the most strongly coupled and the most difficult to derive, also have the simplest matrix form

$$M_q \ddot{q} = Q_q \tag{3}$$

For simplicity, (3) will be referred to as Kane's equations (KE), although there is much more to his method than implied by this equation[5].

To numerically integrate the variables in the above equations, they must be isolated using some form of matrix factorization, followed by forward elimination and back substitution[6]. The matrix in (1) is generally large and sparsely populated with nonzero entries, so specially designed factorization algorithms, which manipulate only the nonzero entries, have been developed[7]. Sparse matrix manipulation algorithms have considerable overhead and are not always reliable. Appending constraints as in (1) and (2) results in systems of differential-algebraic equations with dependent variables, which creates additional numerical and stability problems[8]. Vehicle simulations based on (1) are slow and the results may be unpredictable if a user lacks sufficient numerical analysis background.

Computer programs based on (2) and (3) cover a broad spectrum of solution strategies. Many use full matrix L-U factorization because the coefficient matrices are densely populated with nonzero entries. More recent algorithms use symbolic procedures to generate computer programs which give the factors directly[9-13]. However, effective parallelism is limited by the so-called serial recursion operations which

are required to evaluate the equations and compute the L-U factors. Vehicle simulations based on (2) and (3) tend to be more efficient than those based on (1), but the models may be more difficult to set up.

NEE are attractive because defining vehicle models and formulating equations in Cartesian coordinates is systematic. However, the use of appended constraint equations in (1) that do not contain joint variables leads to numerical difficulties that cannot be fixed by any amount of preprocessing. As noted above, it is difficult to define and develop vehicle models in (2) and (3), but it is virtually impossible to transform (1) into forms equivalent to (3) that can be effectively processed.

3. AUGMENTED NEWTON-EULER EQUATIONS OF MOTION

We use a modified augmented *NEE* approach (*ANEE*) that contains Cartesian coordinates and joint variables to develop equations suitable for direct symbolic reduction to the *KE* form[12,13]. The *ANEE* approach takes the general form

$$\begin{bmatrix} M & C_a^T & 0 \\ C_a & 0 & -H_aB_a \\ 0 & -B_a^TH_a^T & 0 \end{bmatrix} \begin{bmatrix} a \\ -f_a \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} g + C_c^Tf_c - \dot{M}v \\ (\dot{H}_aB_a + H_a\dot{B}_a)\dot{q} \\ B_a^TQ_a \end{bmatrix} \tag{4}$$

which is described below.

A vehicle model starts with a collection of rigid bodies interconnected by idealized nondeforming joints. Its topology, specified by a spanning tree, indicates how the bodies are related by the joints. An invertible Boolean connectivity matrix C_a defines how bodies and joints are related in the spanning tree, and a second Boolean matrix C_c defines how branches are connected by additional chord joints to form closed kinematic circuits. Let $R_a = C_a^{-1}$ and $R_c = -C_cR_a$ define the respective matrix inverses.

Spatial algebra is used to combine rotational and translational quantities into single entities. Spatial rotational and translational operators, which are explicit functions of the joint variables, are used to transform these quantities as needed. Let joint i be defined by a spatial partial velocity h_i where $h_i\dot{p}_i$ gives the relative spatial velocity across the joint and $\dot{p}_i$ is the time derivative of the joint variable(s). If

$$H_a = diag\lfloor h_1, h_2, \ldots, h_{n_a}\rfloor\,;\; H_c = diag\lfloor h_1, h_2, \ldots, h_{n_c}\rfloor \tag{5}$$

$$\dot{p}_a = \left[\dot{p}_1^T, \dot{p}_2^T, \ldots, \dot{p}_{n_a}^T\right]^T;\; \dot{p}_c = \left[\dot{p}_1^T, \dot{p}_2^T, \ldots, \dot{p}_{n_c}^T\right]^T \tag{6}$$

$$v = \left[v_1^T, v_2^T, \ldots, v_{n_a}^T\right]^T \text{ then} \tag{7}$$

$$C_a v = H_a \dot{p}_a \,;\; C_c v = H_c \dot{p}_c \tag{8}$$

where v contains the absolute spatial velocity of every body in the system, and n_a and n_c indicate the number of spanning tree joints and chord joints, respectively.

Solve the first part of (8) for v, and substitute into the second to obtain the loop closure condition

$$R_c H_a \dot{p}_a + H_c \dot{p}_c = 0 \tag{9}$$

which determines joint variable dependency. Equation (9) represents the time derivative of loop constraint equations, $\Phi(p_a, p_c) = 0$, which must be satisfied for all possible configurations. Let

$$q = \dot{I}_a p_a + \dot{I}_c p_c \tag{10}$$

select independent variables from p_a and p_c where $\dot{I}_a$ and $\dot{I}_c$ are Boolean matrices.

The dependent variables are computed using Newton-Raphson iteration,

$$\begin{bmatrix} R_c H_a & H_c \\ \dot{I}_a & \dot{I}_c \end{bmatrix} \begin{bmatrix} \Delta p_a \\ \Delta p_c \end{bmatrix} = \begin{bmatrix} -\Phi(p_a, p_c) \\ 0 \end{bmatrix} \tag{11}$$

when q is given. In vehicle models, the dependent variables are functions of a small number of elements in q. These quantities are precomputed in a preprocessor at discrete intervals of q and approximated by B-splines[14]. This process avoids the use of (11) during run time.

Introducing partial velocities B_a and B_c such that $\dot{p}_a = B_a \dot{q}$ and $\dot{p}_c = B_c \dot{q}$, and substituting into (9) and into the time derivative of (10), and equating coefficients of the independent variables $\dot{q}$ gives

$$\begin{bmatrix} R_c H_a & H_c \\ \dot{I}_a & \dot{I}_c \end{bmatrix} \begin{bmatrix} B_a \\ B_c \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix} \tag{12}$$

where I is an identity matrix. As above, B_a and B_c are precomputed at discrete intervals of q in (12) and approximated by B-splines for evaluation during run time.

The first part of (8) is differentiated and manipulated to obtain the absolute spatial accelerations

$$a = R_a H_a B_a \ddot{q} + R_a\left(\dot{H}_a B_a + H_a \dot{B}_a\right)\dot{q} \tag{13}$$

The partial velocities of $\dot{H}_a$ and $\dot{B}_a$ are also precomputed as functions of q and approximated by B-splines for direct evaluation during run time.

To obtain the *ANEE*, let

$$M = diag\left|M_1, M_2, ..., M_{n_a}\right| ; p = Mv \tag{14}$$

where M contains the entire vehicle's mass and moments of inertia, and p is the total momentum. Differentiating momentum gives the constrained equations of motion as

$$\dot{p} = Ma + \dot{M}v = C_a^T f_a + C_c^T f_c + g \tag{15}$$

$$\text{where } f_a = \left[f_1^T, f_2^T, ..., f_{n_a}^T\right]^T ; f_c = \left[f_1^T, f_2^T, ..., f_{n_c}^T\right]^T ; g = \left[g_1^T, g_2^T, ..., g_{n_a}^T\right]^T \tag{16}$$

represent reaction forces in the spanning tree joints, reaction forces in the chord joints, and all other forces acting in the system, respectively. The constraint reaction forces are projected onto the joint subspaces as

$$Q_a = H_a^T f_a ; Q_c = H_c^T f_c \tag{17}$$

where Q_a and Q_c represent moments and forces that may be applied in the joints.

Equations (13), (15) and (17) define (4) and may be combined to obtain (3) where

$$M_q = (R_a H_a B_a)^T M R_a H_a B_a ; Q_q = Q_q^g - Q_q^q \tag{18}$$

$$\text{with } Q_q^g = B_a^T Q_a + B_c^T Q_c + (R_a H_a B_a)^T g ; Q_q^q = (R_a H_a B_a)^T \left(\dot{M}v + MR_a\left\{\dot{H}_a B_a + H_a \dot{B}_a\right\}\dot{q}\right) \tag{19}$$

The elements of Q_q^q in (19) may be written in factored form as

$$Q_q^q = 1/2 \sum_i \sum_j Q_{qij}^q(q)\dot{q}_i \dot{q}_j \tag{20}$$

where $Q_{qij}^q(q)$ are precomputed and approximated by B-splines for direct evaluation during run time. Many components of $Q_q^g(q,\dot{q})$ are also precomputed and approximated by B-splines for additional run-time efficiency.

4. SPECIAL METHODS FOR OBTAINING REAL-TIME VEHICLE SIMULATIONS

Highly specialized and ad hoc techniques must be employed to obtain real-time vehicle simulations on moderately-sized computers without degrading model integrity. The method of precomputing coefficients and approximating by B-splines was already discussed. Other not-so-obvious techniques

reported in the literature have promising possibilities. An argument made by a number of researchers in real-time dynamics applications is that some simulation accuracy can be sacrificed without significantly degrading model performance. However, numerical integration stability must be preserved or the results could be meaningless. Stable, fixed-step integration methods have been developed that allow substantially larger time steps without degrading the accuracy of most important state variables[8].

With spatial algebra and joint coordinate formulations, each vehicle's equations of motion are formulated in its own chassis coordinate system. This keeps the entire vehicle's inertia matrix nearly constant for all possible vehicle orientations in space, which means that matrix M_q in (18) remains nearly constant. Holding M_q constant at a nominal value meets the accuracy requirements of most mobility, stability and quality assessment applications. Thus its inverse or factors can be precomputed and stored for use during run time. Nominal values for $Q^q_{qij}(q)$ in (20) are also precomputed and stored. Evaluating and factoring these terms during run time, even when interpolating functions are used, represents a substantial percentage of the computational overhead. Depending on model complexity, holding these quantities constant could further increase execution efficiency by several hundred percent.

The incorporation of kinematic differentials[15] into the vehicle kinematics equations is another procedure that substantially reduces run-time computational overhead without degrading model accuracy. For ease of model development and for graphical display purposes, it is generally convenient or necessary to incorporate many bodies into a vehicle model. Using precomputed kinematics and dynamics quantities, most of these bodies will make insignificant contributions to the dynamic solution, so they can be omitted from the real-time simulation algorithms. Kinematic differentials provide a means to do this by accurately filling in the gaps left by the omitted bodies. In a preprocessor, the essential bodies to be retained in the dynamics model are first identified. Then accurate relative displacements between the retained bodies and their kinematic differentials (partial velocities) are computed as functions of the independent variables in q, and approximated by B-splines. However, M_q and $Q^q_{qij}(q)$ are still precomputed with all bodies retained in the model. This process avoids computing the displacements, velocities and accelerations of the omitted bodies during run-time. However, they can all be evaluated later in post processors as functions of the recorded independent variables q and their derivatives.

Using empirically-based subsystem models is another procedure which shows promise for reducing computational overhead without degrading model resolution. A particular real-time simulation model may be practical for evaluating the performance of only a small number of subsystems within a vehicle. And some of these systems may actually be functioning hardware and software which have been interfaced directly with the model. Other complex subsystems such as the engine and transmission may be best represented by empirically-determined performance curves which can be stored in lookup tables and incorporated into the model. For example, the M2-A2 Bradley Fighting Vehicle uses an on-board computer algorithm to control various engine, transmission and steering states. This computer algorithm is installed directly into the model and interfaced with empirical representations of the engine, transmission and steering. Representative track and tire/soil interaction models are also essential for making many vehicle performance predictions, but their model implementations are complex and computationally intensive. Empirically-determined track and tire/soil interaction curves can be used to provide similar inputs to the model at a fraction of the computational overhead.

The state variables used in vehicle models are roughly divided into those with low frequency components and large amplitudes, and those with low and high frequency components but with small amplitudes. For real-time applications, it is important to accurately integrate the first set of variables and approximate the remaining low frequency/small amplitude components. If the high frequency components are important, then real-time simulations are impractical and other procedures must be taken. Stable implicit integration methods are being investigated to accomplish this task[16].

Impact and other transient events within a vehicle model can create havoc with fixed-step integration algorithms. An event may be missed or inaccurately represented, or a simulation could become unstable. A practical method for handling impact events in real-time simulation environments is being studied for incorporation into the models[17]. Consider a jounce stop and suppose it has been isolated from the rest of the vehicle. The state of the jounce stop depends on a small number of vehicle states and loads which are each bounded. In a preprocessor, the isolated high frequency jounce stop equations of motion are integrated at very small time steps over a wide range of initial starting conditions and loads for the duration of one much larger real-time integration time step. Multi-dimensional lookup tables of the

suspension's states at the end of the simulation interval are generated as functions of the initial conditions and assumed loads. During run time at the same integration time step and when a suspension impact is eminent, the suspension's states and interface forces are input to the lookup tables to predict what it will be doing after the step has been taken. Experiments with an M1 dynamic tank model have shown good results[17].

5. NUMERICAL EXAMPLES

Figure 1 shows a computer-generated image of the High Mobility Multipurpose Wheeled Vehicle (HMMWV) and Fig. 2 shows the Bradley Fighting Vehicle M2. These two vehicles are widely different in their functionality and construction. The HMMWV suspension and steering kinematic models are represented by 17 kinematic loops with many dependent variables, and the M2 model contains no loops or dependent variables. The HMMWV model has 16 degrees of freedom and the M2 has 37.

The HMMWV model contains 46 rigid bodies—chassis, 20 suspension elements, four wheels, eight steering elements and 13 drive train elements, including the engine. These elements are interconnected by numerous revolute, universal, spherical and cylindrical joints. An empirically-based drive train model extending from the engine to the half-shafts provides input propulsion to the wheels. Nonlinear tire/terrain interaction models develop vertical, lateral and fore-aft forces to support and propel the vehicle. Kinematic differentials are used to eliminate all bodies but the chassis and four wheels from the dynamic model. This process tremendously reduces computational overhead because only the chassis and wheel displacements and velocities must be evaluated. In this real-time model, B-spline lookup tables are used to evaluate the necessary kinematic quantities for computing dependent displacements and velocities, and to evaluate force inputs from suspensions and steering. Lookup tables are also used to implement the empirical drive train models. Operator inputs to the model are through throttle, braking, steering and gear selection commands. Graphical dynamic displays are from the driver's position or from a birds-eye view.

The M2 model is composed of 32 rigid bodies—chassis, 12 road arms, 12 road wheels, two drive sprockets, two idlers, turret, gun and engine. The bodies are interconnected by revolute joints. The tracks are represented as elastic bands wrapped around the sprockets, idlers and road wheels, which allow the

vehicle to be propelled and steered through the sprockets. Nonlinear wheel/terrain interaction models develop vertical, lateral and fore-aft forces to support and propel the vehicle. An empirically-based drive train model extending from the engine to the drive sprockets, including the electronic controller algorithm taken directly from the vehicle, provides the input propulsion and steering to the model. In this real-time model, lookup tables are required primarily to implement the empirical models because no dependent quantities have to be evaluated. Operator inputs to the model are through throttle, braking, steering and gear selection commands. Graphical dynamic displays are from the driver's position or from a birds-eye view.

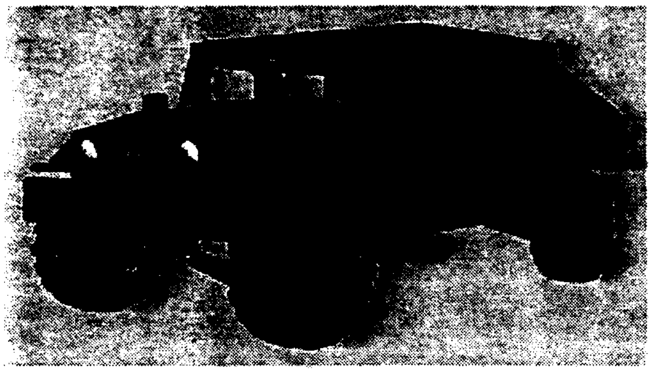

Figure 1

Computer Generated Image of HMMWV

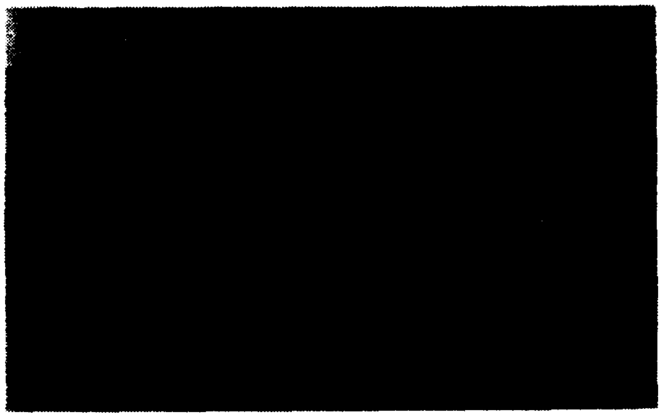

Figure 2

Computer Generated Image of Bradley M2-A2

6. BIBLIOGRAPHY

[1] *Electronic Analog and Hybrid Computers:* G.A. Korn and T.M. Korn, (McGraw-Hill, Inc., New York, NY, 1964).

[2] *A Treatise on Analytical Dynamics:* L.A. Pars, (Ox Bow Press, Woodbridge, CT, 1979).

[3] *A History of Mechanics:* R. Dugas, (London, 1957); English Translation of *Historie de Mécanique*, (Neuchatel, 1950).

[4] *Multibody Systems Handbook:* W. Schielen, (Springer-Verlag, New York, NY, 1990).

[5] *Dynamics: Theory and Applications:* T.R. Kane and D. A. Levinson, (McGraw-Hill, New York, NY, 1985).

[6] *Computer Aided Analysis and Optimization of Mechanical System Dynamics:* E.J. Haug, Editor, (Springer-Verlag, New York, NY, 1984).

[7]*Direct Methods for Sparse Matrices:* I.S. Duff, A.M. Erisman and J.K. Reid, (Clarendon Press, Oxford, 1986).

[8]*Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations:* K.E. Brenan, S.L. Campbell and L.R. Petzold, (North-Holland, New York, 1989).

[9]*Symbolic Computer Methods to Automatically Formulate Vehicle Simulation Codes:* M.W. Sayers, Ph.D. dissertation, The University of Michigan, (1990).

[10]*High Performance Multibody Simulations via Symbolic Equation Manipulation and Kane's Method*: D.E. Rosenthal and M.A. Sherman, *The Journal of the Astronautical Sciences*, Vol. 34, No. 3, (1986), pp. 223-239.

[11]*Symbolic Uncoupling and Efficient Solution of Tree Structured Linear Equation Systems:* R.A. Wehage, *Transactions of the Seventh Army Conference on Applied Mathematics and Computing*, (1990).

[12]*High Resolution Vehicle Simulations Using Precomputed Coefficients:* R.A. Wehage and M.J. Belczynski, ASME *Transportation Systems*, DSC-Vol. 44, (1992), pp. 311-325.

[13]*Constrained multibody Dynamics:* R.A. Wehage and M.J. Belczynski, *Proceedings of the NATO-Advanced Study Institute on Computer Aided Analysis of Rigid and Flexible Mechanical Systems*, Vol. 1, (1994), pp. 461-487.

[14]*Computational Geometry—Curve and Surface Modeling:* B. Su and D. Liu, (Academic Press, Inc. 1989).

[15]*Equations of Motion of Complex Multibody Systems Using Kinematical Differentials:* M. Hiller and A. Kecskemethy, *Transactions of the Canadian Society of Mechanical Engineers*, Vol. 13(4), (1989), pp. 113-121.

[16]*Computational Challenges in Mechanical Systems Simulation:* L.R. Petzold, *Proceedings of the NATO-Advanced Study Institute on Computer Aided Analysis of Rigid and Flexible Mechanical Systems*, Vol. 1, (1994), pp. 195-209.

[17] *The Use of Function Generation in the Real-Time Simulation of Stiff Systems:* R.M. Howe and K.C. Lin, *Proc. of the AIAA Flight Simulation Technologies*, (1990), pp. 217-224.